Алгоритм имитации отжига

Данилов Роман
421 гр.

1 ноября 2024 г.

1 Формальная постановка задачи

Дано n независимых работ из множества $W = \{W_1, W_2, \dots, W_n\}$. Независимость подразумевает независимость по данным. Также определено множество $M = \{M_1, M_2, \dots, M_k\}$ из k процессоров, по которым необходимо распределить работы. И определена функция *complexity*, отображающая множество работ во множество натуральных чисел и отражающая вычислительную сложность конкретной работы.

$$complexity : W \to \mathbb{N}$$

В реализуемой программе входные данные, задающие функцию *complexity* и количество процессоров, представляют собой число k - число процессоров для построения расписания - на первой строке и последовательность N натуральных чисел, разделенных пробелом, на второй:

$$Input$$
$$k \in \mathbb{N},\, k < 100000$$
$$c_i,\, c_i \in \mathbb{N},\, c_i < 1000,\, i = \overline{1, N}$$

, где c_i задаёт вычислительную сложность i-ой работы

Назовём расписанием T следующую двойку:

$$T = (T_1,\, T_2), \text{где}$$
$$T_1 : W \to M,$$
$$T_i = \{\phi_1, \phi_2, \dots, \phi_k\}, \text{где}$$
$$\phi_j - \text{индекс работы, выполняемый на процессоре}\, M_i, j = \overline{1, k}$$

Функция T_1 отвечает за «привязку» работы к процессору. В реализации эта функция будет моделироваться массивом из n элементов, где i-ый элемент хранит номер процессора, на котором выполняется i-ая работа. T_2 - это множество отношений строгого порядка для каждого процессора. Рассмотрим процессор $M_i \in M$. Тогда для работ $W_j, W_k \in W$, таких что $T_1(W_j) = T_1(W_k) = M_i$, W_j выполняется раньше W_k, если $W_j \phi_i W_k$.

В реализации отношение порядка для конкретного процессора будет моделироваться массивом длины равной количеству работ, запланированных на выполнение на этот процессор, где i-ый элемент содержит индекс работы, которая выполняется i-ой по порядку.

Определим функцию старта работы $start_i$ на процессоре M_i, определенную на множестве $I_i = \{W_k \,|\, T_1(W_k) = M_i\}$

$$start_i : I_i \to \mathbb{N}$$
$$start_i(w) = \sum_{k \in I_i \wedge \phi_k < \phi_w} complexity(k)$$

Работа $w \in W$, $T_1(w) = M_i$ выполняется без прерываний, то есть время её завершения всегда равно $start_i(w) + complexity(w)$

Расписание корректно, если для каждой работы задано её распределение на процессор M_i и выполнение работ на процессорах не пересекается. То есть:

$$\forall x \in M \, \exists \, T_1(x),$$
$$\neg \exists x \in W (\exists a \in W (x \neq a \wedge T_1(x) = T_1(a) = M_i \wedge T_2(x) \leq T_2(a) \wedge$$
$$\wedge start_i(a) < start_i(x) + complexity(x)))$$

Необходимо для входных данных построить корректное расписание, минимизирующее суммарное время ожидания. То есть

$$\min_{correct\, T} \sum_{w \in W \wedge T_1(w) = M_i} (start_i(w) + complexity(w))$$